PROGRAMMABLE MANAGEMENT IO PADS FOR AN INTEGRATED CIRCUIT

Inventors: Hoang T. Tran
Howard A. Baumer

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 60/421,780, filed October 29, 2002, entitled "Multi-Rate, Multi-Port, Gigabit Serdes Transceiver," incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The present invention generally relates to serializer/deserializer integrated circuits with multiple high-speed data ports, and more particularly to a serializer and deserializer chip that includes the functionality to switch between multiple high-speed data ports.

Related Art

[0003] High-speed data links transmit data from one location to another over transmission lines. These data links can include serializer data links (i.e., SERDES) that receive data in a parallel format and convert the data to a serial format for high-speed transmission, and deserializer data links (i.e., SERDES) that receive data in a serial format and convert the data to a parallel format. SERDES data links can be used for communicating data through a backplane in a communications system (e.g., Tyco Backplane 16 or 30-inch trace).

[0004] In a high-speed back plane configuration, it is often desirable to switch between multiple SERDES links. In other words, it is often desirable to switch between any one of multiple SERDES links to another SERDES link, and to do so in a low power configuration on a single integrated circuit.

SUMMARY OF THE INVENTION

[0005] A multi-port SERDES transceiver includes multiple parallel ports and serial ports, and includes the flexibility to connect any one of the parallel ports to another parallel port or to a serial port, or both. Furthermore, the multi-port transceiver chip can connect any one of the serial ports to another serial port or to one of the parallel ports. Each parallel port and each serial port includes a plurality of input-output (IO) pads. According to embodiments of the present invention, the pads are programmable to support multiple different electrical specifications, data protocols, timing protocols, input-output functions, and the like.

[0006] The IO pads for the parallel ports are programmable to support different data protocols, including, but not limited to, the XGMII protocol, the Ten Bit Interface (TBI) protocol, the Reduced TBI (RTBI) protocol, and the like. The IO pads are also programmable to support different electrical specifications, including, but not limited to, the High Speed Transistor Logic (HSTL) electrical specification, the Solid State Track Link (SSTL) electrical specification, the Low Voltage Transistor – Transistor Logic (LVTTL) specification, and the like.

[0007] The multi-port transceiver of the present invention is also programmable to support multiple electrical specifications. The transceiver includes a plurality of management data input/output (MDIO) pads. Each MDIO pad is programmable to configure itself and its associated IO pads to comply with the appropriate electrical requirements and data protocols. The electrical specifications and data protocols include IEEE 802.3™ clause 45, IEEE 802.3™ clause 22, or the like.

[0008] Depending on the specified electrical specification and the specified data protocol, the transceiver may be required to support different electrical requirements at the MDIO pad and the adjacent IO pads. Therefore, the MDIO pad is configured to have a separate power connection from the power connection to associated IO pads. In an embodiment, a split-voltage bus

structure is provided to connect the pads for the transceiver to a bus. The structure breaks the power bus VDDO I/O supply, which allows the MDIO pads and the IO pads to operate at different voltage at a given time.

[0009] The multi-port SERDES transceiver also includes a packet bit error rate tester (BERT). The packet BERT generates and processes packet test data that can be transmitted over any of the serial ports to perform bit error testing. The packet BERT can monitor (or "snoop") between the serial ports. In other words, if data is being transmitted from one serial port to another serial port, the packet BERT can capture and store a portion of this data for bit error testing.

[0010] The substrate layout of the multi-port SERDES transceiver chip is configured so that the parallel ports and the serial ports are on the outer perimeter of the substrate. A logic core is at the center of the substrate, where the logic core operates the serial and parallel data ports, and a bus that connects the data ports. The bus can be described as a "ring" structure (or donut "structure") around the logic core, and is configured between the logic core and the data ports. The ring structure of the bus provides efficient communication between the logic core and the various data ports.

[0011] Further features and advantages of the present invention, as well as the structure and operation of various embodiments of the present invention, are described in detail below with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS/FIGURES

[0012] The accompanying drawings, which are incorporated herein and form part of the specification, illustrate the present invention and, together with the description, further serve to explain the principles of the invention and to enable a person skilled in the pertinent art(s) to make and use the invention. In the drawings, like reference numbers indicate identical or functionally similar

elements. Additionally, the leftmost digit(s) of a reference number identifies the drawing in which the reference number first appears.

[0013] FIG. 1 illustrates a multi-port SERDES transceiver chip according to an embodiment of the present invention.

[0014] FIG. 2 illustrates a substrate layout of a multi-port SERDES transceiver chip according to an embodiment of the present invention.

[0015] FIG. 3 illustrates sections of a bus on a multi-port SERDES transceiver chip according to an embodiment of the present invention.

[0016] FIG. 4 illustrates path lengths of wires in a bus on a transceiver chip according to an embodiment of the present invention.

[0017] FIG. 5 illustrates path lengths of wires in a bus on a transceiver chip according to another embodiment of the present invention.

[0018] FIG. 6 illustrates a substrate layout of the multi-port SERDES transceiver chip according to another embodiment of the present invention.

[0019] FIG. 7 illustrates a control system for programming a transceiver pad according to an embodiment of the present invention.

[0020] FIG. 8 illustrates a pad timing controller according to an embodiment of the present invention.

[0021] FIG. 9 illustrates a power bus connection for a multi-port SERDES transceiver chip according to an embodiment of the present invention.

[0022] FIG. 10 illustrates an operational flow for configuring a transceiver pad to support a specified data protocol according to an embodiment of the present invention.

[0023] FIG. 11 illustrates an operational flow for reconfiguring an output transceiver pad to function as an input according to an embodiment of the present invention.

[0024] FIG. 12 illustrates an operational flow for programming a transceiver pad to perform Iddq testing according to an embodiment of the present invention.

[0025] FIG. 13 illustrates an operational flow for changing a timing protocol for a transceiver pad according to an embodiment of the present invention.

[0026] FIG. 14 illustrates an operational flow for configuring a transceiver pad to comply with a specified electrical specification according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0027] FIG. 1 illustrates a multi-port SERDES transceiver 100 according to embodiments of the present invention. The SERDES transceiver 100 includes multiple parallel ports and serial ports, and includes the flexibility to connect any one of the parallel ports to another parallel port or to a serial port, or both. Furthermore, the multi-port transceiver chip 100 can connect any one of the serial ports to another serial port or to one of the parallel ports.

[0028] More specifically, the SERDES transceiver chip 100 includes two parallel transceiver ports 102a-102b and four serial transceiver ports 104a-104d. Other configurations having a different number of ports could be used. The parallel transceiver ports 102a-102b transmit and receive data in a parallel format. The parallel transceiver ports 102a-102b can be XGMII parallel ports, for example, where the XGMII transceiver protocol is known to those skilled in the relevant art(s). Each XGMII port 102 can includes 72 pins, for example, operating at 1/10 the data rate of the serial ports 104.

[0029] The four serial ports 104a-d can be XAUI serial ports, and transmit and receive data in a serial format. Each serial port 104 can be a quad serial port having four serial data lines using the XAUI protocol that is known to those skilled in the relevant art(s). In embodiments of the invention, the serial ports 104 can operate at data rates of 3.125 GHz, 2.5 GHz, and 1.25 GHz. In other words, transceiver chip 100 is a multi-rate device. However, the XAUI data rates above are effectively quadrupled since there are four serial data lines in each serial port 104. The serial ports 104 can be further described as a 10-Gigabit extension sub-layer (XGXS). In embodiments, the serial data ports 104 are differential.

[0030] The parallel ports 102 and the serial ports 104 are linked together by a bus 106. The bus 106 enables data to travel between all the ports 102 and 104. More specifically, the bus 106 enables data to travel from one parallel port 102 to another parallel port 102, and to travel from one parallel port 102 to a serial port 104. Multiplexes 108 connect the bus 106 to the parallel ports 102 and to the serial ports 104. The serial port 104 performs a parallel to serial conversion when receiving parallel data that is to be sent out serial. Likewise, the bus 106 enables data to travel from one serial port 104 to another serial port 104, and to travel between a serial port 104 and a parallel port 102. The parallel port 102 performs a serial-to-parallel conversion when receiving serial data that is to be sent out in parallel. The multi-port SERDES transceiver 100 is highly flexible in being able to connect multiple parallel ports 102 to multiple serial ports 104, and vice versa.

[0031] In embodiments, the SERDES transceiver chip 100 can be implemented on a single CMOS substrate. For example, the SERDES transceiver chip 100 can be implemented using a low power 0.13-micron CMOS process technology, which lends itself to higher levels of integration and application.

[0032] The transceiver 100 enables dual unit operation, where one parallel port 102 is paired up with two of the serial ports 104, and the other parallel port 102 is paired up with the other two serial ports 104. For example, parallel port 102a can be paired with serial ports 104a and 104b. Likewise, the parallel port 102b can be paired with serial ports 104c and 104d. However, there is complete selectivity of the ports that are grouped together for dual unit operation. For example, parallel port 102a can be paired with either serial ports 104a and 104b, or serial ports 104c and 104d. In a backplane configuration, this provides flexibility to connect a parallel port to one or more serial ports for redundancy.

[0033] The transceiver 100 also includes a packet bit error rate tester (BERT) 112. The packet BERT 112 generates and processes packet test data that can be transmitted over any of the serial ports 104 or parallel ports 102 to perform

bit error testing. Any type of packet data can be generated to perform the testing and at different data rates. For example, the packet BERT 112 can generate packet data that can be used to test the SERDES data link. As such, the packet BERT 112 provides a built-in self test for the SERDES data link. The packet BERT 112 generates test data that is sent over one or more of the serial ports 104 using the bus 106 to perform the bit error rate testing of the SERDES data link. Likewise, the packet BERT 112 can capture test data received over any one of the serial ports 104 or parallel ports 102 using the bus 106 for comparison with test data that was sent out. A bit error rate can then be determined based on this comparison.

[0034] In one embodiment, the packet BERT 112 is RAM-based so that the test data is stored and compared in a RAM memory to perform the bit error rate test. In another embodiment, the packet BERT 112 is logic-based so that the test data is generated by a logic function, and transmitted across a SERDES link. Upon receipt back, the test data is re-generated by the logic packet BERT 112, for comparison with the original test data that was sent over the SERDES data link. A RAM packet BERT 112 is more flexible than a logic packet BERT 112 because there is no limitation on the data that can be stored in the RAM packet BERT 112. However, a logic packet BERT 112 is more efficient in terms of substrate area because a RAM occupies more area than a logic circuit.

[0035] Since the packet BERT 112 shares the same bus 106 with the serial ports 104, the packet BERT 112 can monitor (or "snoop") between the serial ports 104. In other words, if data is being transmitted from one serial port 104 to another serial port 104, the packet BERT can capture and store a portion of this data for bit error testing. In one embodiment, the packet BERT 112 "blindly" captures data being sent from one serial port 104 to another serial port 104. In another embodiment, the packet BERT 112 starts capturing data after a particular byte of data is transmitted. In another embodiment, the packet BERT 112 starts capturing data after an error event occurs.

[0036] The SERDES transceiver chip 100 also includes the ability to include other optional logic blocks 114 that are not necessary for the operation of the SERDES transceiver. In other words, these could be customer-driven logic blocks or some other type of logic block. These optional logic blocks 114 can transmit and receive data over the serial ports 104 or parallel ports 102 using the bus 106. The packet BERT 112 and the optional blocks 114 connect to the bus 106 using the multiplexers 110.

[0037] The SERDES transceiver chip 100 also includes a management interface 116 that enables the configuration of the portions (parallel ports 102, serial port 104, packet BERT 112, and optional logic blocks 114) of the transceiver chip 100. In an embodiment, the management interface 116 is configured to be compatible with both IEEE 802.3™ clause 45 and the IEEE 802.3™ clause 22 management standards. The management interface 116 includes two pads 117 that enable two different management chips to program and control the portions of the transceiver chip 100. For example, one management chip connected to pad 117a could control the parallel port 102a and the serial ports 104a and 104b, and another management chip connected to pad 117b could control the parallel port 102b and the serial ports 104c and 104d. The quantity of pads 117 and management chips are provided for illustrative purposes. A greater or smaller quantity of pads 117 and management chips can be included as determined by the system designer.

[0038] FIG. 2 illustrates the substrate layout 200 for the SERDES transceiver 100 according to embodiments of the present invention. The substrate layout 200 is configured to minimize the substrate area of the transceiver 100, and efficiently provide the port interconnections described above.

[0039] The substrate layout 200 is configured so that the parallel ports 102a-102b and the serial ports 104a-104d are on the outer perimeter of the substrate layout 200, as shown. A logic core 202 is at the center of the substrate layout 200, where the logic core 202 operates the bus 106, serial ports 104, and parallel 102 ports. The management interface 116, the packet BERT 112, and the optional logic blocks 114a-114c are adjacent to the logic core 202 as

shown. The bus 106 can be described as a "ring" structure (or donut "structure") around the logic core 202, and placed in between the logic core 202 and the parallel ports 102 and serial ports 104 that occupy the parameter of the substrate layout 200. Furthermore, the ring structure of the bus 106 also provides efficient communication between the logic core 202 and the various ports 102 and 104. Furthermore, the ring structure of the bus 106 also provides efficient communication between the management interface 116, the packet BERT 112, the optional logic blocks 114, and the various ports 102 and 104.

[0040] The bus 106 is illustrated as eight sections 106a-106g for ease of illustration. Each section provides an interface to the respective ports 102 or 104 that are adjacent to the respective sections.

[0041] FIG. 3 represents one of the eight sections 106a-106g of the bus 106 according to embodiments of the present invention. Each section of the bus 106 can be represented as two paths 308 and 310. Data enters the bus 106 through a buffer 302 and proceeds to its destination along the path 308 and through the buffers 304. Once on the bus 106, data passes from one section to another section of the bus 106 using the path 310 and passing through the buffers 312. The mux 306 represents data passing from the bus 106 to a functional block, such as a parallel port 102, serial port 104, or packet BERT 112. The actual wires and buffers on the bus 106 are matched to minimize signal distortion.

[0042] In embodiments, the data wires in the bus 106 are deposited on the substrate for substrate layout 200 in a particular fashion. Namely, a power or ground is placed between adjacent (or near by) data wires. Furthermore, adjacent data wires on the bus 106 are placed on two separate layers. Therefore, a power or ground will be above or below a data wire, and adjacent to a data wire. Therefore, two nearby data wires will not be located directly adjacent to one another, but instead will be positioned diagonally to each other, thereby reducing cross-talk.

[0043] FIG. 4 further illustrates an example layout of the bus 106. The wires 402 between parallel ports 102 and serial ports 104 are configured to have the

same path lengths. In other words, wires 402a-d are deposited so as to have the same path length so as to reduce signal distortion.

[0044] FIG. 5 illustrates another embodiment of the bus 106 in the substrate layout 200. Whereas FIG. 4 depicted only four wires 402a-402d for connecting one port (102 or 104) to an adjacent port (102 or 104), FIG. 5 depicts a plurality of wires 402a-402n for connecting two adjacent ports (102 and 104). The total number of wires 402a-402n is determined by the design of the chip 100.

[0045] In an embodiment, multi-port SERDES transceiver 100 is programmable to support different data protocols, including, but not limited to, the XGMII protocol, the Ten Bit Interface (TBI) protocol, the Reduced TBI (RTBI) protocol, and the like. Transceiver 100 is also programmable to support different electrical specifications, including, but not limited to, the High Speed Transistor Logic (HSTL) electrical specification, the Solid State Track Link (SSTL) electrical specification, the Low Voltage Transistor – Transistor Logic (LVTTL) electrical specification, and the like. The present invention includes methodologies or techniques for sending control signals to configure the parallel ports 102a-102b to support a designated data protocol. This can be explained with reference to FIG. 6, which illustrates a substrate layout 600 for the SERDES transceiver 100 according to another embodiment of the present invention. Substrate layout 600 includes a plurality of pads 604a-604d that are part of the four serial ports 104a-104d. In other words, each serial port 104 includes a plurality of pads 604. As shown, serial port 104a includes a plurality of pads 604a. Serial port 104b includes a plurality of pads 604b. Serial port 104c includes a plurality of pads 604c. Serial port 104d includes a plurality of pads 604d.

[0046] Substrate layout 600 also includes a plurality of pads 602a-602d representing two parallel ports 102a-102b. Pads 602a-602b are part of parallel port 102a, and pads 602c-602d are part of parallel port 102b. Pads 602a and pads 602d are input pads. As such, transceiver 100 receives data and control signals at input pads 602a and input pads 602d. Pads 602b and 602c are

output pads that enable transceiver 100 to transmit data and control signals. In an embodiment, each group of pads 602 includes forty-four individual pads. Forty of the pads are dedicated to sending or receiving data signals, and four of the pads are dedicated to sending or receiving control signals (e.g., clock signals). The total quantity of pads can be increased or decreased as determined by the system designer. Likewise, the ratio of data-to-control pads can also be increased or decreased to meet system requirements as determined by the designer.

[0047] Substrate layout 600 also includes a plurality of management data input/output (MDIO) pads 606a-606d. MDIO pads 606a-606d represent another embodiment of pads 117a-117b, which are described above with reference to FIG. 1. MDIO pads 606a-606d are programmable to configure pads 602a-602d and 604a-604d for compliance with a designated electrical specification and/or data protocol. The electrical specification and/or data protocol is configured via an external pull-up or pull-down resistor(s) at the designated control pad. The electrical specifications include IEEE 802.3™ clause 45, IEEE 802.3™ clause 22, or the like. As shown, MDIO pads 606a control pads 602a-602b, MDIO pads 606b control pads 604c-604d, MDIO pads 606c control pads 602c-602d, and MDIO pads 606d control pads 604a-604b. As discussed above with reference to FIG. 1, in an embodiment, MDIO pads 606 receive instructions from one or more management chips. These instructions are executed by the MDIO pads 606 to configure transceiver 100 and parallel ports 102a-102b to be compatible with the designated electrical specification. As discussed, in an embodiment, one management chip is provided to instruct all MDIO pads 606 and their associated IO pads 602 and/or 604. In another embodiment, a distinct management chip is provided to instruct each MDIO pad 606 and its associated IO pads 602 and/or 604. In another embodiment, a separate management chip is provided to instruct a subset of MDIO pads 606 and their associated IO pads 602 and/or 604.

[0048] The serial IO pads 604a-604d, parallel IO pads 602a-602d, and MDIO pads 606a-606d are positioned to provide rotational symmetry for substrate

layout 600. Therefore, if the transceiver 100 is rotated 180 degrees, the serial and parallel ports can be connected to another communications device without impeding the performance of transceiver 100, or having to reconfigure either device. The symmetrical layout of the components also allows efficiencies to be gained when the transceiver is being connected. For instance, while wire-bonding the pads (i.e., 604a-604d, 602a-602d, and 606a-606d), a technician only needs to design or configure equipment to wire-bond half of the transceiver 100 since the other half would have identical dimensions.

[0049] As discussed, the pads 602a-602d for the parallel ports 102a-102b are programmable to support multiple different standards, protocols, and/or functions. FIG. 7 illustrates a block diagram for logic or circuitry for a control system 700 for programming each pad 602 according to an embodiment of the present invention. Control system 700 includes one or more programmable control registers 702, a pad timing controller 704, input controller 706, output controller 708, and configuration control logic 710. Configuration control logic 710 is responsive to various control signals, which are executed to program pad 602 such that it is capable of supporting a designated protocol. Input controller 706 sends an input control signal 722 to configuration control logic 710 to program pad 602 to receive input. Output controller 708 sends an output control signal 724 to configuration control logic 710 to program pad 602 to send output.

[0050] Control registers 702 includes five types of control signals for programming pad 602. A system operator inputs these control signals, but in an embodiment, the control signals are supplied by a computer system (not shown). The five control signals include a reset message 712, an Iddq message 714, a power down message 716, a pad type message 718, and a delay select message 720.

[0051] A reset message 712 is released to instruct pad 602 to change its originally designated function (i.e., input or output). For example, if pad 602 is originally designated as an output pad, the pad 602 is reconfigured to operate as an input pad upon receipt of a reset message 712. In FIG. 7, pad

602 is an output pad. Therefore, reset message 712 is only delivered to input controller 706 to enable pad 602 to switch to receiving input.

[0052] An Iddq message 714 is released to implement Iddq testing to measure the quiescent supply current of transceiver 100. When executed, Iddq message 714 places the path across a pad 602 in a quiescent state to measure the leakage current. As shown, Iddq message 714 is sent to input controller 706, output controller 708, and/or configuration control logic 710 for implementation.

[0053] A power down message 714 is released to suspend the operations of portions of pad 602. If power down message 714 is delivered to input controller 706, pad 602 no longer receives input. If power down message is delivered to output controller 708, pad 602 no longer outputs data or control messages. If power down message 714 is delivered to configuration control logic 710, the muxing and timing operations of the control logic 710 are suspended.

[0054] PAD type message 718 specifies the data protocol and electrical specification, and instructs configuration control logic 710 to implement the specified data protocol and electrical specification. As discussed, the data protocol includes the XGMII, TBI, RTBI protocols, and the like. The electrical specification includes HSTL, SSTL, and LVTTL electrical specifications, and the like.

[0055] Delay select message 720 specifies the path delay for input and output. The parameter specified in the delay select message 720 enables the system operator, or the like, to adjust the delay between input and output at each pad 602 for better system performance.

[0056] As discussed above, the present invention allows transceiver 100 to be programmed to support different data protocols. Referring to FIG. 10, flowchart 1000 represents the general operational flow for configuring a programmable pad 602 to support a designated data protocol, according to an embodiment of the present invention.

[0057] The control flow of flowchart 1000 begins at step 1001 and passes immediately to step 1003. At step 1003, protocol instructions for a designated data protocol are specified. Referring back to FIG. 7, the specified protocol instructions are placed in programmable control registers 702.

[0058] At step 1006, a control signal carrying the protocol instructions are released to program a pad 602. Referring back to FIG. 7, the control signal is shown as PAD type message 718, which is received by configuration control logic 710.

[0059] At step 1009, the control signal (i.e., PAD type message 718) is executed to implement the specified data protocol. At step 1012, an output control signal 724 or input control signal 722 is sent to configuration control logic 710 to instruct the programmable pad 602 to function as an output or input. At step 1015, pad 602 transmits or receives in accordance with the specified data protocol. Afterwards, the control flow ends as indicated at step 1095.

[0060] Referring back to FIG. 7, pad 602 is programmed to function as an output. However, pad 602 can be reconfigured to function as an input. Referring to FIG. 11, flowchart 1100 provides an example of a general operational flow for reconfiguring an output programmable pad 602 to function as an input.

[0061] The control flow of flowchart 1100 begins at step 1101 and passes immediately to step 1103. At step 1103, pad 602 is instructed to cease functioning as an output. Referring back to FIG. 7, power down message 716 is sent to output controller 708, which as a result, stops sending output control signal 724.

[0062] At step 1106, input operations are initiated at pad 602. Referring back to FIG. 7, reset message 712 is sent to input controller 706 to initiate the operations. At step 1109, input control signal 722 is sent to configuration control logic 710. At step 1112, configuration control logic 710 executes the input control signal 722 to configure pad 602 to start receiving input. Afterwards, the control flow ends as indicated at step 1195.

[0063] As discussed above, programmable control registers 702 also release an Iddq message 714 to implement Iddq testing. Referring to FIG. 12, flowchart 1200 provides an example of a general operational flow for programming pad 602 to perform Iddq testing.

[0064] The control flow of flowchart 1200 begins at step 1201 and passes immediately to step 1203. At step 1203, Iddq message 714 is released to either input controller 706 or output controller 708, depending on the I/O operations currently designated for pad 602. At step 1206, Iddq message 714 is also released to configuration control logic 710, which programs pad 602 to measure leakage as previously discussed. Afterwards, the control flow ends as indicated at step 1295.

[0065] As shown, if pad 602 is operating as an input pad, pad timing controller 704 receives pad data 726 from pad 602. The delay select message 720 instructs pad timing controller 704 to buffer the pad data 726 for a prescribed time period before sending the data to its destination as internal data 728. The prescribed time period is substantially the same as the path delay at other pads 602.

[0066] Conversely, if pad 602 is operating as an output pad, pad timing controller 704 receives internal data 728 and buffers the data for a prescribed time period before enabling it to be output as pad data 726.

[0067] FIG. 8 represents the buffering process for implementing path delay according to an embodiment of the present invention. As shown, pad timing controller 704 includes a plurality of buffers 802a-802n and a multiplexer 804. Data enters pad timing controller 704 and is delayed in one or more buffers 802a-802n for a prescribed time period. The incoming data can be pad data 726 received by pad 602, or internal data 728 received from another portion of transceiver 100.

[0068] Each buffer 802a-802n delays the incoming data a fixed delay time. The delay time is fixed internally. In other words, the system designer specifies the delay time for the buffers during fabrication of transceiver 100, and this value is not changed by the control registers 702 or a system operator.

The data is sent to the next buffer 802a-802n unless multiplexer 804 opens the communications path to receive the data. The delay select message 720 determines when multiplexer 804 opens the communications path. The communications path can be opened prior to the data entering one of the buffers 802a-802n, or at any point after the data is released from one of the buffers 802a-802n. Therefore, the delay select message 720 enables the path delay to be increased or decreased by specifying the number of buffers 802a-802n, if any, that the data should traverse. Once the data is received by multiplexer 804, the data is sent to its destination as pad data 726 or internal data 728.

[0069] Hence, the multi-port SERDES transceiver 100 includes the ability to change the timing of parallel ports 102 and serial ports 104. This includes the ability to change the timing between the data and clock signals. In other words, the registers in the parallel ports 102 and serial ports 104 can be re-programmed to operate at different timing protocols. Referring to FIG. 13, flowchart 1300 provides an example of a general operational flow for changing the timing protocol for a pad 602.

[0070] The control flow of flowchart 1300 begins at step 1301 and passes immediately to step 1303. At step 1303, one or more parameters are input to adjust the path delay. Referring back to FIG. 7, the parameters are entered at programmable control registers 702.

[0071] At step 1306, the delay parameters are communicated to PAD timing controller 704. Referring back to FIG. 7, the delay parameters are encoded in delay select message 720.

[0072] At step 1309, the delay parameters (i.e., delay select message 720) are executed to specify the total delay period for the path delay. As discussed with reference to FIG. 8, the total delay period is measured by the quantity of buffers 802a-802n that data must traverse before being received by multiplexer 804.

[0073] At step 1312, data (i.e., PAD data 726 or internal data 728) is received, and at step 1315, the data is delayed the specified total delay period. At step

1318, the data is sent to its destination. Afterwards, the control flow ends as indicated at step 1395.

[0074] As discussed with reference to FIG. 6, each MDIO pad 606a-606d is programmable to configure itself to comply with a designated electrical standard, such as IEEE 802.3™ clause 22, IEEE 802.3™ clause 45, or the like. For instance, IEEE 802.3™ clause 22 specifies the access to management scheme, including data protocol and electrical requirements.

[0075] Pads 602a-602d are programmable to support any combination of data protocols (e.g., XGMII, TBI, RTBI, etc.) and electrical specifications (e.g., HSTL, SSTL, LVTTL, etc.), and the electrical requirements are determined by the designated electrical specification. For example, the SSTL electrical specification requires pads 602a-602d to operate at 2.5 volts. The HSTL electrical specification requires pads 602a-602d to operate at 1.5 volts or 1.8 volts. The LVTTL electrical specification requires pads 602a-602d to operate at 2.5 volts or 3.3 volts.

[0076] Notwithstanding the electrical requirements for pads 602a-602d, MDIO pads 606a-606d must operate at 1.2 volts to comply with IEEE 802.3™ clause 45. To comply with IEEE 802.3™ clause 22, MDIO pads 606a-606d must operate at 2.5 volts. Accordingly, MDIO pads 606a-606d are programmable to configure themselves and their associated pads 602a-602d to comply with the appropriate electrical requirements. For example, to comply with IEEE 802.3™ clause 45, the power connection to the MDIO pads (e.g., pads 606c) and their corresponding input and output pads (e.g., 602d and 602c) must be broken to allow the MDIO pads to operate at 1.2 volts and the input/output pads to operate at 2.5 volts. To enable the split voltage requirement to be implemented, a split-voltage bus structure is provided to connect the pads for transceiver 100 to a bus. An embodiment of a split-voltage bus structure is illustrated in FIG. 9.

[0077] FIG. 9 illustrates power supply connections for MDIO pads 606c-606d and output pads 602c, according to an embodiment of the present invention. The power supply connections include VDDO I/O supply 912, VSSO I/O

supply 914, VSSC core supply 916, and VDDC core supply 918. MDIO pads 606c-606d are separated from output pads 602c by split voltage structure 902a-902b. Structure 902a-902b breaks the power bus VDDO I/O supply 912, which allows different electrical requirements to be provided for MDIO pads 606c-606d and the adjacent output pads 602c. Hence, the power signals 904, data signals 906, clock signals 908, and ground signals 910 for MDIO pads 606c-606d will not interfere with the electrical and data signals communicated from output pads 602c. The connection for the VSSO I/O supply 914, VSSC core supply 916, and VDDC core supply 918 is not broken by the structure 902a-902b.

[0078] Referring to FIG. 14, flowchart 1400 provides an example of a general operational flow for configuring a programmable pad (i.e., serial IO pads 604a-604d, parallel IO pads 602a-602d, and MDIO pads 606a-606d) to comply with a specified electrical standard, such as IEEE 802.3™ clause 22, IEEE 802.3™ clause 45, or the like.

[0079] The control flow of flowchart 1400 begins at step 1401 and passes immediately to step 1403. At step 1403, MDIO instructions are accessed to identify the specified electrical specification (e.g., HHTL, SSTL, LVTTL, etc.). As discussed, the MDIO pad 606 must operate at a certain voltage, depending on the specified electrical specification.

[0080] At step 1406, the MDIO instructions are executed to configure the electrical requirements for the associated IO pads 602 and/or 604. As discussed, the IO pads 602 and/or 604 may be required to operate at a different voltage than the MDIO pad 606.

[0081] Once the electrical requirements have been configured, the control passes to step 1409. At step 1409, data and control signals are sent or received at the MDIO pad 606 and IO pads 602 and/or 604 in accordance with the specified electrical specification. Afterwards, the control flow ends as indicated at step 1495.

Conclusion

[0082] Example embodiments of the methods, systems, and components of the present invention have been described herein. As noted elsewhere, these example embodiments have been described for illustrative purposes only, and are not limiting. Other embodiments are possible and are covered by the invention. Such other embodiments will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.